Training Spaces

Fostering machine sensibility for spatial assemblages through wave function collapse and reinforcement learning

Alessandro Mintrone[1], *Alessio Erioli*[2]
[1,2] *University of Bologna*
[1] *alessandro.mintrone5@gmail.com* [2] *alessio.erioli@unibo.it*

This research explores the integration of Deep Reinforcement Learning (RL) and a Wave Function Collapse (WFC) algorithm for a goal-driven, open-ended generation of architectural spaces. Our approach binds RL to a distributed network of decisions, unfolding through three key steps: the definition of a set of architectural components (tiles) and their connectivity rules, the selection of the tile placement location, which is determined by the WFC, and the choice of which tile to place, which is performed by RL. The act of thinking becomes granular and embedded in an iterative process, distributed among human and non-human cognitions, which constantly negotiate their agency and authorial status. Tools become active agents capable of developing their own sensibility while controlling specific spatial conditions. Establishing an interdependency with the human, that engenders the design patterns and becomes an indispensable prerequisite for the exploration of the generated design space, exceeding human or machinic reach alone.

Keywords: *Reinforcement Learning, Machine Learning, Proximal Policy Optimization, Assemblages, Wave Function Collapse*

INTRODUCTION

As AI simultaneously pervades and restructures our technological ecology, it also reshapes our cognitive processes and habits along with it. Conquering territories of knowledge that were thought to be inviolably human only few years ago, complex non-human cognitions compel us to rethink our model of authorship, acknowledging the implied intricacies, and questioning both our relationship with tools and conception of creativity.

We propose to explore a paradigm for architectural design in which non-human forms of cognitions are embedded into the decision network (Clark 2008). Instead of mere passive devices, we consider those forms active agents able to promote, select, reinforce and inhibit design directions by their own affordances (Leach 2016). These liberated tools, instead of being aimed at parroting human thinking, are acknowledged in their own sensibility and biases; non-human cognitions enabled to claim a broader autonomy and authorship coparticipation, by continuously negotiating both their agency and their authorial status (Picon 2016) (Parisi 2014).

General Adversarial Network (GAN) based applications of AI for the generation of architectural proposals such as Bolojan [1], Chaillou (2020), del Campo (2019, 2020) train algorithms to produce outputs as wholes, mostly in the form of images, be they plans or pictures, out of other images or language. These applications leverage on an idea of conception deliberately declared as a form of intuition, in which AI replace in some measure the human mind in the act of conceiving a fully-fledged outcome. Instead, we employ Reinforcement Learning (RL) to develop a local distributed behavior, which continuously mediates between internal and external conditions, and whose outcome over time is a three-dimensional assemblage, generated by an inherently spatial and material-aware process.

In order to form three-dimensional spatial organizations, RL must be coupled with an iterative generation algorithm such as Wave Function Collapse (WFC), which relies on a discrete representation of both space and connectivity structure [2]. Given a limited set of parts (tiles), their local connection rules, and information about the topological structure of space, the algorithm can unravel a vast array of different spatial conditions. Albeit the WFC design space is ripe with variety, the algorithm nature makes it fragmented and its unaided navigation (i.e. converging towards an established goal by tweaking initial conditions) impossible.

We present a methodology for steering the generation of these assemblages towards specific spatial qualities via a continuously enacted feedback-loop between the human designer and the AI. Coupled with WFC, an Artificial Neural Network (ANN) is trained implementing Proximal Policy Optimization (PPO) (Schulman et al. 2017 [6]). Thus, the system takes responsibility in shaping the global space by learning how to perform local component selection, ostering the development of a cognitive structure capable of pursuing specific and articulated spatial conditions resulting from the iterative assemblage of three-dimensional parts. More specifically, our results show how, selecting quantified spatial descriptors representing both local and global features, it is possible to characterize the assemblage's spatial qualities, enabling the designer's analysis and intervention, while providing continuous feedback to the algorithm.

METHODS

Assemblages

The notion of assemblages we refer to is Manuel DeLanda's expansion of Deleuze's *agencements* (Deleuze and Guattari 1987): arrangements in which both the qualities of parts and their mutual relations play a crucial role in defining the qualities of the whole (DeLanda 2006). DeLanda aims to move beyond the structuralist metaphor of the organism (parts have no existence outside the whole) without returning to the collage model (there are no relations, only individual parts), since both show a limited ability to explain emergence (Johnson 2001). In the assemblage framework, while parts maintain their own individual identity, they might acquire further characterization from their interactions inside the assemblage. Also, they can be detached from an assemblage and plugged in another one where, while maintaining their embedded "properties", they can exhibit different relational "capacities", afforded by the mutated interactions within the new assemblage.

DeLanda's theory is aimed at society at large, and establishes a theoretical framework based on parts and their mutual relations, independent of the application domain and its specific nature. In architecture, focusing on their topological (and not semantic or structural) aspects, assemblages align with a view based on tectonics (the construction from parts, proceeding by addition or growth), rather than hylomorphism (the imposition of a figure over inert matter) or stereometry.

The design of parts, or components, and their connectivity, plays a key role in the design process. In a typical functionalist approach, the elementary component is a finalized unit, its function not supposed to exceed the designated purpose; a column or a beam correspond to single, specific elements. To fully inquire the assemblages' potential, it is necessary to loosen up this specificity, leaving space for genericity and incompleteness: a radical opening to foster context-dependent emerging conditions, as well as differentiation and variety, all based on the constitutive interrelation among parts. As a consequence, the definitions and understanding of architectural semantics and tectonics change: "column" or "beam" rather than elements, identify conditions that can be expressed by one as well as several elements, while each element can determine multiple conditions at once.

The design process we propose unfolds through three key moments in which critical choices are deployed: the definition of a set of tiles, their connectivity rules, and the grid of cells to be populated, the selection of the next location where a tile will be placed, which is determined by the WFC algorithm, and the choice of which tile to place, which is performed by the ANN. While tiles and connectivity design happen outside the simulation, the system affords mutual feedback-loops of influence: tiles and connectivity design shape the assemblage's space of possible configurations, while the assessment of the assemblage qualities reveals the tiles capacities and provides the necessary input to act back on the tiles themselves.

Wave Function Collapse

Wave Function Collapse (WFC) is a constraint solving algorithm, that iteratively places tiles in a predetermined grid of cells, complying with a set of provided adjacency rules, regardless of the tiles content (Figure 1). In particular the algorithm has gained momentum in game design, especially for procedural textures synthesis and world generation, since it spatializes information in a coherent topological structure, generating aperiodic patterns. Our implementation, made in *Unity3D*, is intended both as a platform for the generation and exploration of a large field of three-dimensional assemblages, and as training environment for an ANN, making goal-oriented spatial generation manageable via Machine Learning techniques.

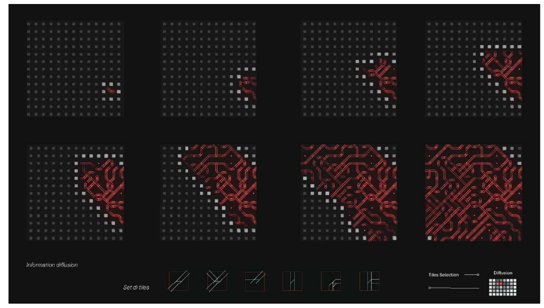

Figure 1
Iterative tile placemente inside the grid.

WFC requires a discretized representation of space, a grid defined by a set of cells along with the topological structure of their connections. The algorithm is not limited by a regular grid or a specific network topology; different grids have been explored, both in two and in three dimensions, ranging from square to hexagonal, up to space-filling polyhedral grids such as cubical, rhombic dodecahedral and truncated octahedral. The initial implementation was performed on a bidimensional square grid in which every cell is connected with its adjacent ring of eight neighbors (four sides and four corners). This setup combines an affordable simplicity in tiles design with a rich, yet manageable, connectivity set, while granting complex enough outcomes and clarity in the assessment phase.

Each cell is initialized to an unobserved state; instead of containing a tile, the state is the superposition of all the probabilities of containing every specific tile (akin to a picture resulting from the semi-transparent overlay of several images). A single iteration consists of three phases: *observation, collapse* and *propagation*. During each observation, a new unobserved cell is selected and collapsed to a defined state containing a particular tile. WFC relies on an entropy function to measure, for every cell, the degree of uncertainty about the possible tiles it may contain. The next cell to collapse is selected choosing the cell with the lowest entropy value; in other words, the point with the lowest level of uncertainty, which contains the lowest number of possible states. This fosters more coherence into the assemblages, making the algorithm less prone to fall into contradiction (i.e., the impossibility to place a tile that satisfies all the connectivity constraints). To quantify uncertainty, we follow the original implementation by Maxim Gumin [2] evaluating the Shannon entropy. Given a discrete random variable x, if $x_i, \ldots, x_n$ are the possible outcomes, and $P(x_i), \ldots, P(x_n)$ their probabilities to occur, the entropy of x is defined as:

$$H(x) = -\sum_{i=1}^{n} P(x_i) \cdot \log P(x_i) \quad (1)$$

Subsequently, during the collapse phase, a tile is placed, randomly selecting among those who satisfy the connectivity constraints. Then, during propagation, the new information gained from the previous collapse is diffused. For each neighbor cell the list of suitable tiles and the entropy values are recursively updated based on the new constraints defined by the placed tile. If no tile satisfies these constraints, the algorithm falls into contradiction and stops, otherwise the process continues with another iteration, until grid completion [3].

Tiles Design and Spatial Configurations

Every tile is provided with three types of information: a set of connection rules determining the allowed adjacencies in every direction, the contained geometry, and, in order to establish all its possible permutations, a symmetry type identifier. A large number of tilesets was tested on both bidimensional and three-dimensional grids. Complexity brews quickly: even a slim tileset grants the emergence of a large collection of articulated and diverse outcomes. This abundance in results largely depends on the careful design of the adjacency constraints; minor changes can mark the difference between a successful system and one prone to contradiction.

The algorithm manifests a tendency towards pattern homogeneity and self-similarity in its spatial outcomes (Figure 2). Though those may be desired qualities, said results still represent a narrow subset of all the spatial configuration the components are inherently able to produce. The random tiles selection mechanism gives each tile the same probability to occur without the possibility to control this choice; as a consequence, the system lacks the capacity to stabilize patterns outside its bias range and/or orient the final outcome. To address this limitation, some authors assign probability "weights" to each tile [4], affecting their chances of being selected. We propose to train an ANN leveraging Deep Reinforcement Learning techniques to perform this choice. The aim is to achieve, by mean of an intelligent, context-aware and open-ended agency, a wider range of configurations with more control on their spatial qualities.

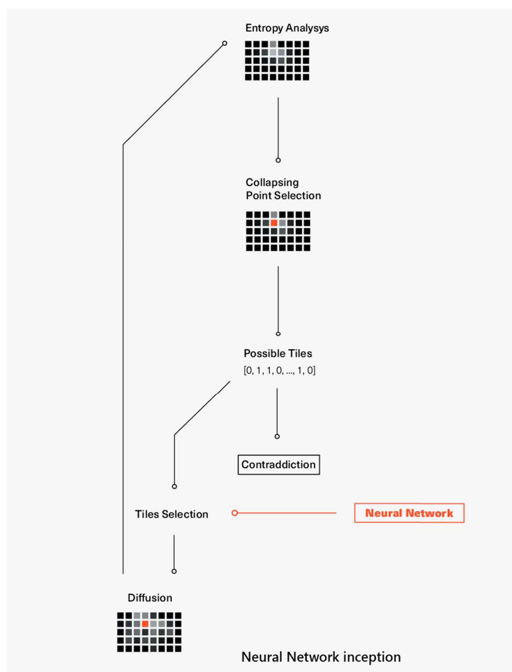

Figure 2
Algorithm structure and AI insertion

Reinforcement Learning

Differently from other kinds of Deep Learning algorithms (Goodfellow 2001), RL is not based on a pre-existing dataset (Figure 3). The ANN instead learns its behavior from the experience accumulated by an agent interacting with an environment, trying to maximize a series of rewards awarded after its actions. In this process, the algorithm itself generates its own set of data from which its behavior is developed. The agent and the environment are part of a mutual feedback-loop. The agent performs actions based on its perception of the environment's current state. These actions modify the environment, changing its state and, by doing so, affecting the agent's subsequent actions.

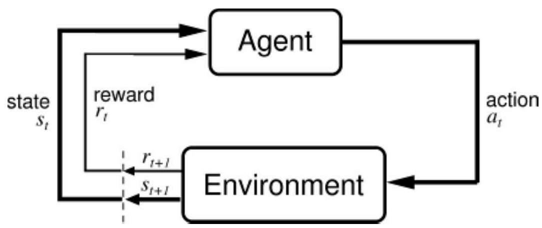

Figure 3
Information flow in RL.

During the training process, the environment provides an additional signal: a positive or negative reward associated to every state that guides the training process. The agent shapes its behavior trying to maximize this reward, learning from how its actions lead to higher or lower values. Therefore, a careful engineering of the reward function is essential to orient and widen the spectrum of agent's behaviors. If defining such function can be straightforward for simple problems, when complexity increases or when facing open-ended problems such as the ones characterizing architecture, translating the desired qualities of the outcome in terms of rewards becomes challenging.

Nonetheless, since the designer is required to set goals instead of hardcoding behaviors inside the algorithm, a task that proves to be hard in terms of human understanding, such as mapping the complex non-linear correlations between local actions and global outcomes, can be externalized to the RL algorithm that trains the ANN and iteratively refines its policy. This method appears more capable of approaching those ill-defined problem whose boundary conditions are difficult to trace, since, instead of learning how to deliver complete, definite solutions, what the algorithm develops is a sensibility, the ability to produce structured yet adaptable proposals, interacting with the environment and navigating different, often conflicting, conditions.

Quantitative Spatial Analysis

A set of six quantified spatial descriptors is defined with a twofold purpose: trying to characterize space by analyzing different qualities at both the global and local scale, and defining the associated reward functions. Translating architectural spatial qualities into quantitative design goals for the ML algorithm, they enable the comparative analysis of the resulting assemblages' features, and set up the data in a computable and intelligible form for the AI.

1. Density. Every tile has an associated local density value, representing the degree to which the cell is filled. Density is defined as the average local density of the observed cells. The associated reward is calculated by defining a *desired density* and then comparing it with the actual one.

$$1 - |\text{desiredDensity} - \text{actualDensity}| \qquad (2)$$

2. Spatial distribution. While density is a global parameter, spatial distribution is defined as the coherence of a desired local density distribution with the actual one. The associated reward is calculated by computing at each cell's collapse the difference between the *desired local density* and the *actual local density*. During training the desired distribution is randomly generated by the system via an attractor field. After the training, the designer can provide a custom desired distribution.

$$1 - |\text{desiredFieldValue} - \text{actualFieldValue}| \qquad (3)$$

3. Orientation. Principal directions are associated to each tile from a predetermined set, in a range of none to two per tile. Orientation designates the main direction inside the assemblage, if the main direction does not surpass the others at least by a threshold amount, the assemblage is marked as *non-directional*. The corresponding positive reward is awarded if the main direction corresponds to the *de-*

sired main direction.

$$\frac{\text{directionalTiles}}{\text{totalTiles}} > \text{threshold} \quad (4)$$

4-5. Structural and spatial connectivity. The tiles are divided into two categories according to their local density value: *void tiles* and *solid tiles*, which when connected with others from the same category form structural and spatial clusters respectively. When working in a tridimensional space, the information about whether or not each structural cluster is connected to the ground is also retained. Setting a threshold, it is possible to identify, for each structural cluster, tiles clusters that are either disconnected from the ground, or whose size is below the threshold.

$$1 - \frac{\text{tilesUnderThreshold}}{\text{totalTiles}} \quad (5)$$

6. Planar connectivity. This parameter is defined only for a three-dimensional assemblage. A plane representing the element orientation can be assigned to each tile by providing its normal vector. Tiles which plane is horizontal (within a given threshold), are clustered as in connectivity analysis. Given a threshold, planar connectivity and its reward are defined with the same formula used for structural and spatial connectivity.

Training

The implemented machine learning algorithm is PPO (Proximal Policy Optimization), a state-of-the-art deep RL class of algorithms developed by *OpenAI* (Schulman et al. 2017 [6]). This model, unlike previous RL algorithms, is able to operate with continuous inputs and outputs. It selects actions relying on an advantage function that estimates the expected value of each possible choice and updates its policy according to the divergence between its inferences and the actual outcome.

The algorithm is implemented in the *Unity ML-Agents Toolkit*, a library seamlessly integrated in *Unity3D* (Juliani et al. 2020 [5]). The ANN architecture consists of 3 hidden -layers, each containing 256 units. This architecture has proved sufficiently robust when operating in 2D and 3D with different sets of tiles, while maintaining the same structure and hyper-parameters.

Given a starting grid of cells, a training episode is made of as many iterations are needed to complete an assemblage or run into a contradiction. At each iteration, the environment provides information about its current state, and the collected inputs are normalized (remapped in a 0-1 range). Information regarding the allowed tiles in the collapsing cell and the tiles already placed in the adjacent ones is represented in one-hot encoder form. The remaining information is given in a continuous form as normalized scalars, and contains both the desired and actual values of the selected spatial features, along with their relative weights.

Subsequently, the ANN selects, among a list of allowed elements, which tile place inside the collapsing cell. The ANN returns an array of normalized values as a one-hot encoder, representing the probability of each candidate tile to be the overall most valuable choice; sampling from this probability distribution, the tile is selected. After updating the environment and awarding the correspondent reward, the ANN policy is also updated.

For each training, a subset of active spatial qualities is set, a weight is provided for every feature, and the corresponding rewards are scaled so that the maximum total achievable reward during each training episode is 1. Additional reward is awarded for each completed step, up to a maximum of 0.25, encouraging the agent to complete the assemblage without falling into contradiction.

During training, a generalization function modifies the desired values and their relative weight at given intervals. Multiple instances of the assembling algorithm, each one with different values, are run at the same time. These two strategies prevent the ANN from overfitting to a limited set of goal values. The training performance is monitored via TensorBoard, assessing through its generated charts the cumulative reward, as well as the episode length and behavior entropy, averaged over the last 100 episodes.

Resulting Assemblages

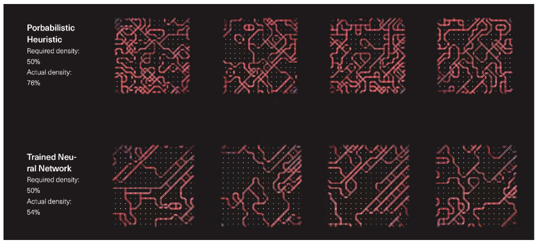

Figure 4
Results: comparison between heuristic guided and AI guided assemblage.

The assemblages realized by the trained ANN are compared against a stochastic baseline that assigns to each tile a different probability in a random choice (Figure 4). In one set of 25 tests, in which the shared goal was to obtain an assemblage with a target density of 50%, the ANN consistently outperformed the baseline (average 54% against 73% of the baseline), while also exhibiting more structured patterns (Figure 5).

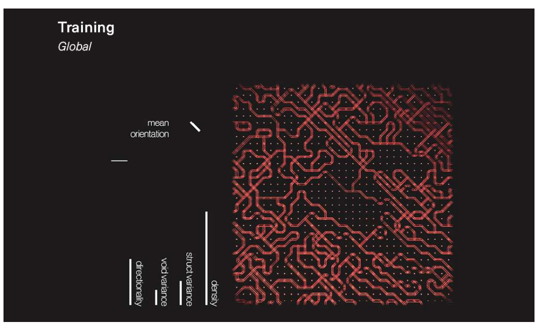

Figure 5
Results: Output of the ANN guided assemblage when trained on all the descriptors at once.

Given a common set of initial tiles, each descriptor consistently led to the appearance of specific patterns in the outcome. Since these patterns emerge as a byproduct of the repetition of certain sequences of actions, they display how strategies developed to maximize the assigned reward also yield related formal qualities. These qualities and their causal correlation with the set goals can be clearly discerned in the case of a single descriptor, while in the case of a larger number of descriptors this correlation is more difficult to discern, hindering the designer's ability to fine tune the training. Despite the inherent difficulties of understanding a system with high dimensionality and interrelated simultaneous parameters, this can be considered a limit of this approach.

The emergent cognitive behavior is not bounded to a defined dimension of the grid, so what is learned in the training environment's limited space can be applied to larger assemblages. This behavior is contingent to the inherent history of the undergone training: repeating the training process, even with the same parameters, can lead to different actions and strategies in the same conditions and with the same reward values. In this sense the descriptors are not intended as objective representations of the assemblage's spatial features, but as stimuli to hone the AI's sensibility.

Mapping and Visualizing

Transitioning from a bidimensional space to a tridimensional one, the increased dimensionality and related number of permutations leads to an inflation of the possible tiles to compute and, consequently, of the number of parameters inputted into the ANN. In order to maintain sufficient design agility and provide a design space that facilitates the individuation of the ANN contribution, after experimenting with sets differentiating in tiles amount, geometry and connectivity, a simple and limited set of tridimensional tiles is adopted: a total of six planar elements, three horizontal, two vertical, and one diagonal (Figure 6). The set was used to produce a database of approximately 10,000 assemblages performed in a 10x10x10 grid containing the placed tiles as well as the coded resulting spatial qualities (Figure 7).

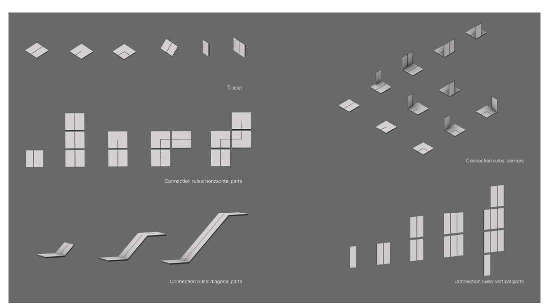

Figure 6
Three-dimensional tileset and connectivity rules.

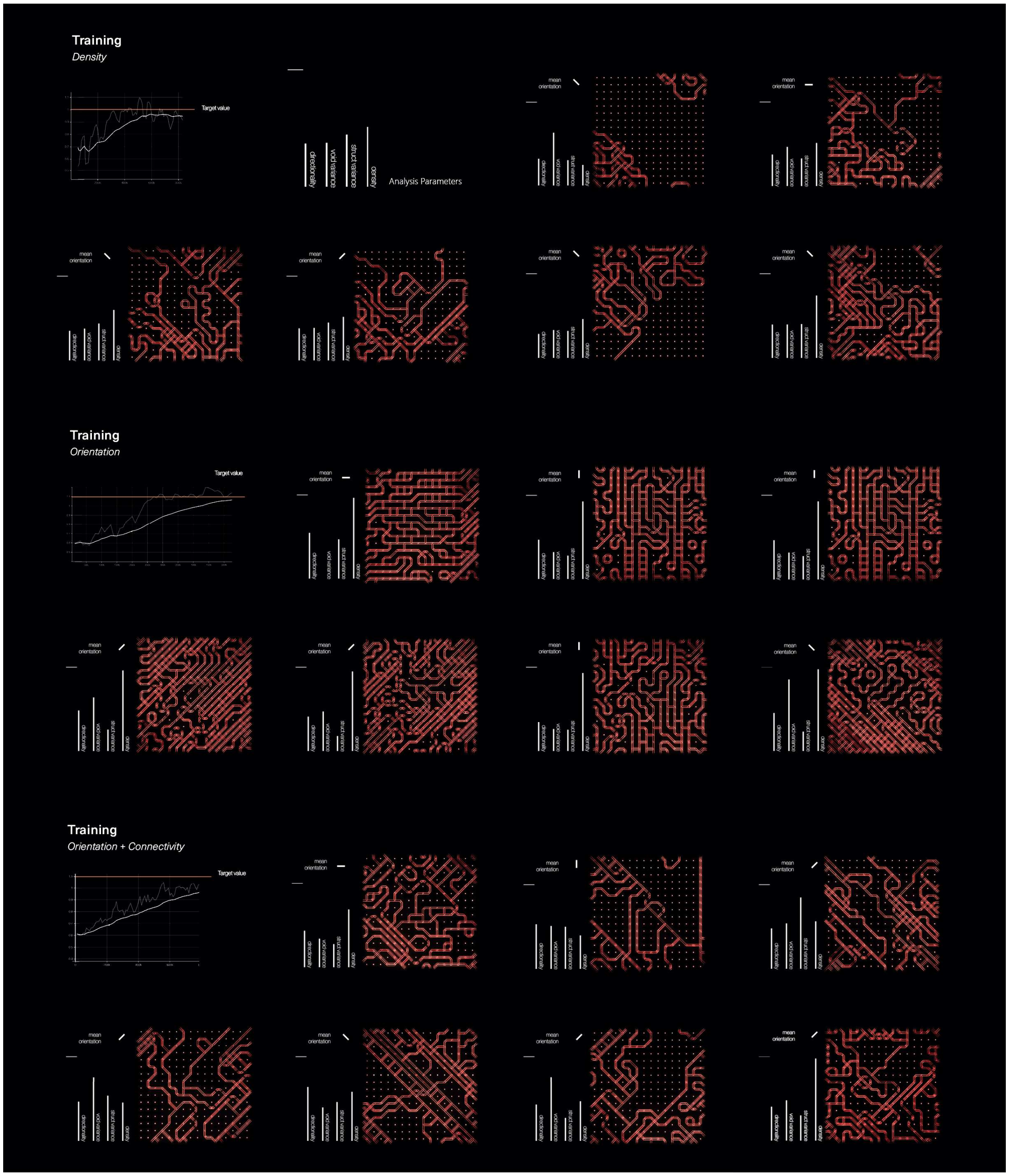

Figure 7
Results: Output of the ANN guided assemblage when trained on density, orientation and a combination of orientation and connectivity.

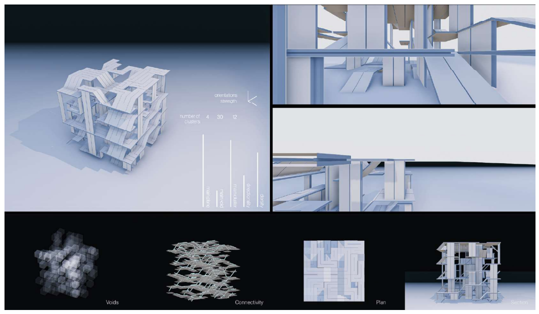

Figure 8
Multiple visualizations of a resulting assemblage.

Superabundance of mathematically unique results is a typical consequence of procedural generative processes, and WFC makes no exception. This excess of information defies analysis in human terms. Manual methods are unfeasible for such large amounts, while quantitative analysis on its own fails to account for all the qualitative and perceptual features embedded in the assemblage. Furthermore, the understanding of a three-dimensional intricate spatial assemblage cannot rely on visual inspection alone: making comparisons, and even grasping the spatial structure of a single episode, is a complex cognitive task which can be significatively eased when supported by an externalized representation (Clark 2008). Therefore, an interface, mediating between human and non-human cognitions and intelligible to both is required to help orientation while navigating the algorithm-generated outcomes. A coordinate system is implemented using the previously established quantitative spatial descriptors, making the database addressable for human and machinic inquiries alike. This orientation system, allows the selection of a specific assemblage based on the analyzed spatial qualities: it is coupled with a visualization system that displays the resulting three-dimensional assemblage, an automated selection of multiple viewpoints, and various graphic representation of the parameters set (Figure 8). Quantitative and qualitative features complement each other, enhancing the completeness of the assessment process.

However, the analysis system is still limited by the difficulty of visualizing and examining this high dimensional space in a three-dimensional representation (Figure 9). Dimensionality reduction techniques such as *Self-Organizing Maps* (Harding 2016), although not implemented at the present stage of research, can help this process.

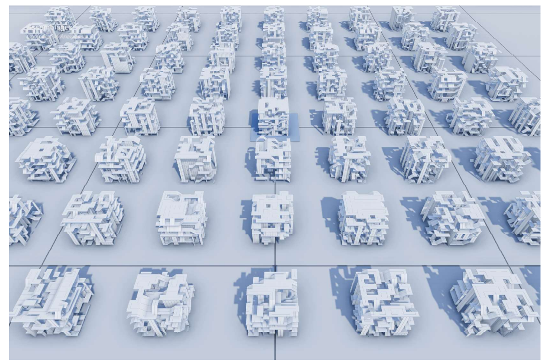

Figure 9
Global view on the resulting assemblages

CONCLUSIONS

This research explores a granular and distributed paradigm for cognition and creation of architectural space. Acknowledging the limitations of a self-contained human creativity and challenging the myth of its primacy, the generation process adopted is not anthropocentric, as it puts the human in a more extended cognitive ecology where the algorithmic intelligence of WFC and ANN participate as co-contributors with autonomous agency and authorship.

The current state of our research refers to a notion of architectural space that is deliberately partial to its topological features and includes only visual accounts of phenomenological qualities. Despite such self-imposed restrictions and the limitations that ensue, quantitative and qualitative complexity must be reckoned with early on. The presented method provides a compass to navigate an otherwise scattered design space: the descriptors perform the dual role of connective tissue for the design space and coordinate system for the analysis of multiple outcomes.

Goals and reward functions are indicators of abstract performances formulated by a human. Given a set of tiles and connectivity constraints, each descriptor is entangled with its own emergent patterns and form stable connections with the outcome's properties. Nonetheless, said qualities and their expression exceed the boundaries of this influence, since they are the result of autonomous machinic sensibilities developed during the training. Thanks to the WFC properties these patterns are extendable, with unaltered tileset and training, to larger or multiple grids, therefore suitable for agile design iterations. Even so, combined goals interact in a non-linear fashion, which defies a priori effect prediction and detection of individual contributions. The use of machine learning techniques to deal with high dimensionality (such as self-organizing maps) seems promising, and it is considered as a future implementation to strengthen post-generation analysis.

The interplay among tile design, WFC, and RL produces topologically coherent spatial structures, which are directly computable based on intrinsic measurable properties and their inherent data representation. Our goal is neither a phenomenological use of AI, where ANNs are trained to develop features based on preexisting catalogs, nor the reduction of the architectural outcome to its fitness against some predetermined condition. Instead, we aim at the expansion of the creative architectural horizon by including non-human, autonomous agency and sensibility in the design of spatial arrangements.

REFERENCES

del Campo, M, Carlson, A and Manninger, S 2020, 'Towards Hallucinating Machines - Designing with Computational Vision', *International Journal of Architectural Computing*, Online, pp. 1-16

del Campo, M and Manninger, S 2019 'Imaginary Plans', *Proceedings of the 2019 ACADIA Conference – Ubiquity and Autonomy*, Austin, pp. 412-418

Chaillou, S 2020 'Archigan: Artificial Intelligence X Architecture', *Architectural Intelligence*, Singapore, pp. 117-127

Clark, A 2008, *Supersizing the Mind: Embodiment, Action, and Cognitive Extension*, Oxford University Press, New York

DeLanda, M 2006, *A New Philosophy of Society: Assemblage Theory and Social Complexity, Annotated edition*, Continuum, London

Deleuze, G and Guattari, F 1987, *A Thousand Plateaus*, Minnesota University Press, Minneapolis

Goodfellow, I, Bengio, Y and Courville, A 2016, *Deep Learning*, MIT Press, Cambridge

Harding, J 2016 'Dimensionality Reduction for Parametric Design Exploration', *Advances in Architectural Geometry 2016*, Zurich, pp. 274-287

Johnson, S 2001, *Emergence: The Connected Lives of Ants, Brains, Cities and Software*, Scribner, New York

Juliani, A, Berges, V P, Teng, E, Cohen, A, Harper, J, Elion, C, Goy, C, Gao, Y, Henry, H, Mattar, M and Lange, D 2020, *Unity: A General Platform for Intelligent Agents*, arXiv:1809.02627v2

Leach, N 2016 'Digital Tool Thinking: Object Oriented Ontology versus New Materialism', *ACADIA 2016 POSTHUMAN FRONTIERS: Data, Designers, and Cognitive Machines Projects Catalog of the 36th Annual Conference of the Association for Computer Aided Design in Architecture*, Ann Arbor, p. 344–351

Parisi, L 2014, 'Automated Architecture: Speculative Reason in the Age of the Algorithm', in Mackay, R and Avanessian, A (eds) 2014, *#Accellerate#, the accelerationist reader*, Urbanomic Media Ltd, Falmouth, pp. 401-424

Picon, A 2016, 'Free the Robots!', *Log*, 36, pp. 146-151

Schulman, J, Wolski, F, Dhariwal, P, Radford, A and Kilmov, O 2017, *Proximal Policy Optimization Algorithms*, arXiv:1707.06347v2

[1] https://vimeo.com/425905084

[2] https://github.com/mxgmn/WaveFunctionCollapse

[3] https://robertheaton.com/2018/12/17/wavefunction-collapse-algorithm/

[4] https://github.com/heyx3/EasyWFC

[5] https://arxiv.org/abs/1809.02627

[6] https://arxiv.org/abs/1707.06347